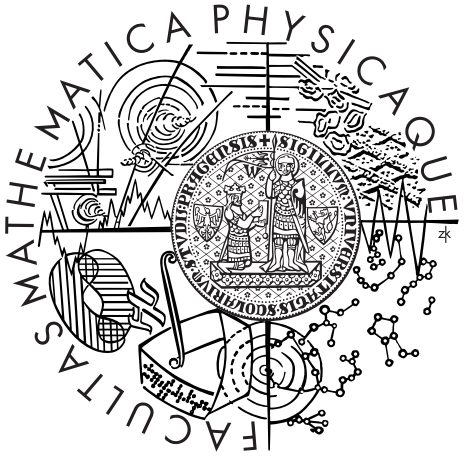
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Ján Pich

Těžké tautologie

Katedra Algebry

Vedoucí diplomové práce: Prof.RNDr. Jan Krajíček, DrSc.
Studijní program: Matematika
Studijní obor: Matematické struktury

Praha 2011

Poděkování

Ďakujem vedúcemu mojej diplomovej práce Janovi Krajíčkovi za cenné konzultácie, rady a podporu. Ďakujem Stefanovi Dantchevovi za starostlivé vedenie mojej práce počas Erasmus pobytu na Durham University (Michaelmas term 2010 a Epiphany term 2011). Ďakujem mojim rodičom za podporu počas celého štúdia.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborný zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ..... Podpis autora

Název práce: Těžké tautologie

Autor: Ján Pich

Katedra: Katedra Algebry

Vedoucí diplomové práce: Prof.RNDr. Jan Krajíček, DrSc.

Abstrakt: Skoumáme nedokazatelnost tvrzení $\mathrm{NP} \not\subseteq \mathrm{P/poly}$ v různých fragmentech aritmetiky. Ta se obvykle dosahuje ukázáním těžkosti výrokových formulí kódujících superpolynomiální spodní odhady pro booleovské obvody.
Nejprve prezentujeme několik známých technik a tvrzení. Přirozené důkazy, efektívní interpolaci, KPT větu, iterovatelnost, gadget generátory atd.
Pak dokážem několik původních výsledků. Ukážeme nedokazatelnost superpolynomiálních spodních odhadů na booleovské obvody v systémech s efektívní interpolací (modulo složitostní předpoklad) a v systémech podobajících se stromovým Frege systémům manipulujícím s formulemi, které obsahují jen málo proměnnych dokazovaného tvrzení.
Tyto výsledky jsou založeny na dokazování těžkosti Nisan-Wigdersonových generátorů v príslušných důkazových systémech.

Klíčové slova: Důkazová složitost, Booleovské obvody, Nisan-Wigderson generátor

Title: Hard tautologies

Author: Ján Pich

Department: Department of Algebra

Supervisor: Prof.RNDr. Jan Krajíček, DrSc.

Abstract: We investigate the unprovability of $\mathrm{NP} \not\subseteq \mathrm{P/poly}$ in various fragments of arithmetic. The unprovability is usually obtained by showing hardness of propositional formulas encoding superpolynomial circuit lower bounds.
Firstly, we discuss few relevant techniques and known theorems. Namely, natural proofs, feasible interpolation, KPT theorem, iterability, gadget generators etc.
Then we prove some original results. We show the unprovability of superpolynomial circuit lower bounds for systems admitting certain forms of feasible interpolation (modulo a hardness assumption) and for systems roughly described as tree-like Frege systems working with formulas using only a small fraction of variables of the statement that is supposed to be proved.
These results are obtained by proving the hardness of the Nisan-Wigderson generators in corresponding proof systems.

Keywords: Proof complexity, Circuit lower bounds, Nisan-Wigderson generators

Contents

Introduction

A potential approach for separating P and NP is to show that there is an NP problem that cannot be solved by polynomial-size boolean circuits. In this thesis we investigate the unprovability of the nonuniform version of P$\neq$NP, that is of NP$\not\subseteq$P/poly.

The most convenient way how to obtain the unprovability of superpolynomial circuit lower bounds in relatively strong mathematical theories like bounded arithmetics is to prove that propositional formulas encoding superpolynomial circuit lower bounds are hard (i.e. do not have short proofs) in corresponding propositional proof systems.

Razborov [29, 31] studied the proof complexity of formulas encoding circuit lower bounds and considered them as good candidate hard formulas for as strong proof systems as Extended Frege (EF).

Using known translations to bounded arithmetic the hardness of superpolynomial circuit lower bounds for EF would imply the unprovability of superpolynomial circuit lower bounds in S^1_2. This theory is a fragment of Peano arithmetic (with induction schema restricted to a subclass of bounded formulas) which captures a lot of contemporary complexity theory.

All in all, an interesting phenomenon occurs here. While the approach to prove the unprovability of superpolynomial circuit lower bounds is, in fact, positive in the sense that it looks for a possibility of the existence of efficient algorithms for hard problems, it is actually realized by looking for potential hard tautologies. That is, even if we obtained the unprovability result it would not have to be clear if this is because there are efficient algorithms for hard problems or because the non existence of such algorithms is demonstrated already by the hardness of these propositional formulas.

In the first chapter we describe in more details the relation between first-order theories of bounded arithmetic and propositional proof systems.

Then we present proof complexity generators used to express circuit lower bounds as propositional formulas and mention some known conditional hardness results based on feasible interpolation and natural proofs.

Subsequently, we discuss techniques like iterability, gadget generators and some general strategies for obtaining hardness of circuit lower bounds.

In the second part of the thesis we present some original results concerning the Nisan-Wigderson (NW) generators. We prove that the NW-generators are hard for proof systems with feasible interpolation [25], and that superpolynomial circuit lower bounds are hard for proof systems with formula interpolation property (assuming the existence of P/poly functions hard for subexponential size formulas).

Next we prove the hardness of the NW-generators and also the hardness of superpolynomial circuit lower bounds for certain systems that might be roughly described as tree-like Frege systems working with formulas depending only on a small fraction of lines of the given generator.

The last part of the thesis is dedicated to a scaled-down version of Krajíček's result from [19]. Namely, we show that a theory corresponding to Frege systems

cannot separate P and NP by proving circuit lower bounds. This result, however, does not yield superpolynomial lower bounds on Frege because we use an unsuitable encoding of formulas. Nevertheless, it is an interesting result with a potential for further improvement.

We assume basic knowledge of computational complexity and mathematical logic.

1 Bounded Arithmetic and Propositional proof systems

In mathematics we usually express our statements and proofs in terms of first-order logic. However, interesting statements like NP$\not\subseteq$P/poly can be often captured in a sequence of propositional formulas.

The existence of short proofs of such sequences of formulas in certain propositional proof systems corresponds to the provability of original first-order statements in certain fragments of arithmetic. This reduces the investigation of predicate logic to the investigation of propositional logic which is often considered as a simpler mathematical concept.

1.1 Propositional proof systems

The formal definition of propositional proof systems we use comes from the seminal paper of Cook and Reckhow [8]. Here, $\{0,1\}^n$ denotes the set of all binary strings of length n, $\{0,1\}^* = \bigcup_n \{0,1\}^n$, $Rng(f)$ denotes the range of a function f and $TAUT$ stands for the set of all tautologies. As tautologies can be naturally encoded into binary strings, we see $TAUT$ as a subset of $\{0,1\}^*$.

Definition 1 (Cook-Reckhow [8]). *A propositional proof system is a poly-time function $P : \{0,1\}^* \mapsto \{0,1\}^*$ such that $Rng(P) = TAUT$. Any x such that $P(x) = y$ is called P-proof of y.*

This complexity theoretic definition allows Cook and Reckhow [8] to show that NP=coNP is equivalent to the existence of the so called p-bounded proof system, a propositional proof system with poly-size proofs of all tautologies.

While the propositional logic might seem to be quite simple, propositional proof systems can be very complex.

Any usual first-order theory T containing some arithmetic can be seen as a propositional proof systems P_T. A proof of a formula in P_T is just a T-proof of the statement formalizing that the formula is tautology. As proofs in any usual theory T are poly-time recognizable, P_T is computable in poly-time.

The converse holds in some sense too. Whenever we have a propositional proof system, we can use it as a part of a formalization of mathematics. Consequently, ZFC or any theory formalizing currently used mathematics, is practicaly the strongest propositional proof system we can obtain.

We are, however, interested in much simpler and more usual proof systems, e.g. Hilbert calculus. This is because for such systems we have the mentioned correspondence between the existence of short proofs and the provability of universal statements in the relevant fragments of arithmetic.

Definition 2 (Cook-Reckhow [8]). *A Frege rule is a $k+1$-tuple of formulas $A_0, ..., A_k$ such that any truth assignment satisfying all formulas $A_0, ..., A_{k-1}$ satisfies also A_k. A Frege rule where $k = 0$ is called a Frege axiom.*

Let F be a finite set of Frege rules. A Frege proof of ϕ from $A_1, ..., A_l$ is a finite sequence $\psi_1, ..., \psi_k$ of formulas such that $\psi_k = \phi$ and each ϕ_i is either one

of $A_1, ..., A_l$ or it is derived from previous formulas by application of a Frege rule from a finite set of Frege rules.

A Frege proof system F is given by any finite set of Frege rules which is sound (every formula with an F-proof is a tautology) and implicationally complete (whenever a set of formulas X entails a formula ϕ, there is an F-derivation of ϕ from X).

The Frege systems satisfy the formal definition of propositional proof systems because Frege-proofs are poly-time recognizable. In each step of the proof one needs to check just a finite set of derivation rules.

If we restrict the rules to work only for formulas of constant depth we obtain a natural restriction of Frege systems, constant depth Frege systems. On the other hand, we can extend the Frege systems in the following way.

Definition 3 (Cook-Reckhow [8])**.** *An Extended Frege system (EF) is any Frege system which can in addition use inference rules of the form*

$$q \equiv B$$

where atom q does not appear in B, it does not appear in any previously derived formulas, nor in the last formula of the proof.

Another well studied proof system is resolution.

Definition 4. *Resolution is a proof system operating with clauses of literals (i.e. disjunctions of variables and their negations) and with one derivation rule which given two clauses $C \vee x$ and $B \vee \neg x$ infers clause $C \vee B$. A resolution proof of a (DNF) formula ϕ is a resolution derivation of empty clause from $\neg\phi$ expressed by a set of clauses (in a CNF form).*

Resolution is much weaker system than Frege. It does not prove PHP (the pigeonhole principle) efficiently [10] while Frege does [5]. On the other hand, it is possible to extend resolution naturally by a rule allowing to "name formulas" by new variables into a system which is as strong as Extended Frege.

Also, while it is known that constant depth Frege does not prove PHP efficiently [1] there are no non-trivial lower bound for Frege systems or for any stronger system.

1.2 Feasible provability

We will now present the corresponding theories of arithmetic and briefly describe theirs relation to the propositional counter parts. The detailed description can be found in [4], [7] or [14].

The corresponding theory for EF is S_2^1. This theory introduced by Buss [4] captures feasible proofs in the sense that the Σ_1^b definable functions in the theory are precisely the poly-time functions. Thus, roughly speaking, intermediate constructions in S_2^1 proofs are feasible.

The language of S_2^1 is $\{0, S, +, \cdot, \#, |x|, \lfloor \frac{1}{2}x \rfloor, =, \leq\}$ where S is the successor function, $|x|$ is the length (of binary representation) of x, and $x\#y = 2^{|x|\cdot|y|}$.

Let sharply bounded quantifiers be those of the form $\forall x \leq |t|$ or $\exists x \leq |t|$ where x does not occur in the term t. The class Σ_1^b consists of the formulas containing only existential bounded quantifiers (like $\exists x \leq t$) and sharply bounded quantifiers of both kinds.

The axioms for S_2^1 consists of universal sentences defining the symbols of $L_{S_2^1}$ together with the following restricted induction scheme:

$$[\phi(0) \wedge \forall x(\phi(\lfloor 1/2x \rfloor) \rightarrow \phi(x))] \rightarrow \forall \phi(x)$$

where $\phi(x)$ is any Σ_1^b formula.

Buss [4] showed that the functions Σ_1^b definable in S_2^1 are precisely the poly-time functions.

Let $A(x)$ be a Σ_0^b formula. That is a formula where only sharply bounded quantifiers are allowed. Then, there is a natural translation of $\forall x A(x)$ into a sequence of propositional formulas $a_1, a_2, ..$ where a_n expresses validity of A on inputs of length n, cf [14]. It is known that such $\forall x A(x)$ theorems of S_2^1 can be translated into propositional formulas with poly-size EF-proofs, cf [7], [14].

To get **the theory corresponding to Frege** it is convenient to go to the two-sorted setting, cf [7].

The two-sorted language L^2 has *number* variables that range over $\mathbf{N}$ (natural numbers), and *string* variables that range over finite subsets of $\mathbf{N}$ (interpreted as finite binary strings). Further, it contains usual functions and predicates for numbers: $0, 1, +, \times, =, \leq$ and set membership $t \in X$ (or simply $X(t)$), set equality $=_2$ and string length $|X|$ which is 0 if X is empty, and $1 + max\{z; z \in X\}$ otherwise. (So $|X|$ is roughly the length of the binary string corresponding to X.)

The class Σ_0^B consists of formulas whose only quantifiers are bounded number quantifiers while free string variables are allowed.

V^0 is axiomatized by defining axioms for symbols in L^2 together with the comprehension axiom scheme for Σ_0^B formulas

$$\exists X \leq y \forall z < y(X(z) \leftrightarrow \phi(z))$$

where $\phi(z)$ is any Σ_0^B formula. Here, $\exists X \leq y\phi$ stands for $\exists X(|X| \leq y \wedge \phi)$.

Let $A(x, X)$ be a Σ_0^B formula. Then if $\forall x \forall X A(x, X)$ is a theorem of V^0, it can be translated into tautologies with poly-size proofs in constant depth Frege.

VNC^1 is an extension of V^0 by an axiom describing a poly-time algorithm evaluating a balanced Boolean formula.

Consider a monotone Boolean formula represented as a binary tree H with $2a - 1$ nodes (a leaves and $a - 1$ inner nodes). The a leaves of H are numbered $a, ..., 2a - 1$, and the two children of an inner node x are $2x$ and $2x + 1$. Each inner node x ($1 \leq x \leq a - 1$) is labeled with either $\wedge$ or $\vee$. Therefore to encode H we need just a string G of length $\leq a$ so that $G(x)$ encodes label of node x of H. Let $G(x)$ hold if and only if node x of H is an $\wedge$-gate. Then define

$$\delta(a, G, I, Y) \equiv \forall x < a\,(Y(x + a) \leftrightarrow I(x)) \wedge [0 < x \Rightarrow$$
$$Y(x) \leftrightarrow [(G(x) \wedge Y(2x) \wedge Y(2x + 1)) \vee (\neg G(x) \wedge (Y(2x) \vee Y(2x + 1))]]$$

VNC^1 is the theory over L^2 which is axiomatized by V^0 and

$$\forall a \forall G \forall I \exists Y \delta(a, G, I, Y)$$

Let $A(x, X)$ be a Σ_0^B formula. Then, if $\forall x \forall X A(x, X)$ is a theorem of VNC^1, it can be translated into tautologies with poly-size proofs in Frege system, cf [7].

1.3 Ajtai's method

Propositional translations of universal theorems have polynomial-size propositional proofs (in suitable theories and propositional systems). The other direction holds in some sense too.

Propositional translations of universal sentences that are not provable (in a specific way expressed in the following theorem) have no poly-size propositional proofs (again, in relevant proof systems). This gives us the so called Ajtai's method [1] for proving lower bounds on propositional proof systems.

Let T_{NC^1} be the true universal theory of $\mathbf{N}$ (natural numbers) in the language L_{NC^1} that consists of all NC^1 functions and relations. According to a certain $RSUV$ isomorphism (see [7]) it can be seen as an extension of VNC^1.

Let M be a non-standard model of true arithmetic in the language L_{NC^1}. Let $n \in M$ be a non-standard number and define M_n to be the substructure of M consisting of numbers whose bit length is less than n^k for some standard $k \in \mathbf{N}$. Note that M_n is closed on NC^1 functions, in particular the following relations are in M_n

$Fla(x, y)$: y encodes formula of poly-size in x

$Prf_{Frege}(x, z, y)$: z encodes Frege-proof of y of size $\leq poly(x)$

$Sat(x, a, y)$: truth assignment a satisfies formula y; $a, y \leq poly(x)$

$Ref_{Frege}(x, z, y, a)$: $(Fla(x, y) \wedge Prf_{Frege}(x, z, y)) \rightarrow Sat(x, a, y))$

Theorem 1 ([20]). *Let T_k be tautologies of size $k^{O(1)}$, for all $k \in \mathbf{N}$. Assume that for an arbitrary choice of non-standard n there is a model N of T_{NC^1} that is a cofinal extension of M_n such that there is a truth assignment $w \in N$ to atoms of T_n that falsifies the formula in N, i.e. $Sat(n, w, \neg T_n)$ is valid in N.*

Then tautologies T_k do not have poly-size Frege proofs.

This theorem can be generalized in various ways. Scaling-up model M_n to consist of elements of subexponential size (as well as bounds on size of formulas Fla, Prf_{Frege},...) would lead to subexponential lower bounds.

More importantly, the theorem holds also for other corresponding theories of bounded arithmetic and propositional proof systems. For example, PV and EF. The crucial thing is that PV is closed on poly-time functions and proves Ref_{EF}.

2 Proof complexity generators

Proof complexity generators were independently introduced by Alekhnovich, Ben-Sasson, Razborov and Wigderson [2] and by Krajíček [16]. They allow us to encode circuit lower bounds as propositional formulas. Moreover, they give us a good framework for exploration of various modifications and generalizations of circuit lower bounds.

Definition 5. *A proof complexity generator* $g : \{0,1\}^* \longmapsto \{0,1\}^*$ *is a function computed by* $m^{O(1)}$*-size circuits* $\{C_n\}$ *representing restrictions of* g, $g_n : \{0,1\}^n \longmapsto \{0,1\}^m$ *for some injective function* $m = m(n) > n$.

For a proof complexity generator g *and any string* $b \in \{0,1\}^m$ *define the* τ*-formula* $\tau(C_n)_b$ *as* $b \not\equiv C_n(x)$. *The variables of* $\tau(C_n)_b$ *are* $x_1, ..., x_n$ *for inputs of* C_n, *and* $y_1, ..., y_{m^{O(1)}}$ *for gates of* C_n.

$\tau(C_n)_b$ is a tautology iff $b \notin Rng(C_n)$. We shall denote the formulas simply $\tau(g)_b$ because circuits C_n are though as canonically determined by g.

Definition 6. *A generator* g *is a hard proof complexity generator for a propositional proof system* P *iff there is no polynomial size* P*-proof of any* $\tau(g)_b$ *(for* m *tending to infinity), i.e. for any sequence* $b_1, b_2, ...$ *(*$|b_1| < |b_2| < ...$*) formulas* $\tau(g)_{b_i}$ *do not have poly-size* P*-proofs.*

In order to prove a superpolynomial lower bound for a proof system P it would be sufficient to prove the hardness of $\tau(g)_b$ just for one sequence of b's but we believe that for generators of interest this τ-formulas are hard for all b's.

Now, using τ-formulas we can encode circuit lower bounds. Firstly, recall that a circuit of size s can be encoded by $O(s \log s)$ bits.

Definition 7. *Let* $s \geq k \geq 1$. *The truth table function* $tt_{s,k}$ *takes as input* $O(s \log s)$ *bits describing a size* $\leq s$ *circuit* C *with* k *inputs, and outputs* 2^k *bits: the truth table of the function computed by* C. $tt_{s,k}$ *is equal to zero string at inputs that do not encode a size* $\leq s$ *circuit with* k *inputs.*

For any boolean function f on k variables represented by its truth table, $2^{O(k)}$-size formulas $\tau(tt_{s,k})_f$ say that f has no s-size circuits. Of course, this makes sense only if s is not ridiculously big, like $2^{O(k)}$. In particular, formulas $\tau(tt_{s,k})_f$, for $s = k^{\omega(1)}$, expressing superpolynomial circuit lower bounds on f are well defined.

It is an interesting question whether one could similarly capture the P$\neq$NP statement. Obviously, we could take care only about hardness of NP functions. However, it is not known how to encode the uniformity of Turing machines into a sequence of propositional formulas.

3 General strategy and known techniques

We want to prove the hardness of circuit lower bounds. Therefore, we can assume that these circuit lower bounds are true. Otherwise, they would be trivially unprovable.

More generally, our intention is to take a fundamental assumption that there is no efficient way how to solve hard problems, like NP$\not\subseteq$P/poly or similar, and use it to show that there is no efficient proof of say NP$\not\subseteq$P/poly in a natural proof system.

Conversely, the crucial task here is to extract a useful computational information from a given proof.

3.1 Natural proofs

The well-known natural proofs barrier by Razborov and Rudich [32] exhibits the same kind of flip: one can use existence of a certain kind of proofs of superpolynomial circuit lower bounds to construct an efficient computational test breaking strong pseudorandom generators.

Informally, a circuit lower bound proof is called P/poly-natural against P/poly (resp. NP/poly-natural against P/poly) if it implies the existence of a set of boolean functions C, s.t.

- any sequence of functions from C is hard for P/poly.
- deciding membership (of functions represented by theirs truth tables) in C is in P/poly (resp. in NP/poly)
- for any n there are at least $2^{2^n-O(n)}$ functions on n inputs in C

Furthermore, let us define a strong pseudorandom generator (SPRNG) as a sequence of functions $G_n : \{0,1\}^n \mapsto \{0,1\}^{2n}$ such that for some $\epsilon > 0$ there are no circuits C of size $S < 2^{n^\epsilon}$ such that

$$|\mathbf{P}_x[C(G_n(x) = 1)] - \mathbf{P}_y[C(y) = 1]| \geq 1/S$$

where x is taken at random from $\{0,1\}^n$, and y is random from $\{0,1\}^{2n}$ (using uniform distribution).

Theorem 2 (Razborov-Rudich [32])**.** *If there exists a SPRNG, then there is no P/poly-natural proof against P/poly.*

Razborov and Rudich [32] also showed that known circuit lower bounds on restricted classes of circuits are natural (this is not true for the recent result of Williams [34] that is based on diagonalization). Thus, theirs result is considered as a strong barrier against possible separation of P and NP.

Rudich [33] later attempted to extend the natural proofs barrier into the context of non-deterministic circuits. He proved that if the so called super-bits exist, then there are no NP/poly-natural properties against P/poly.

Here, super-bit is a sequence of P/poly functions $g_n : \{0,1\}^n \mapsto \{0,1\}^{n+1}$ such that for some $\epsilon > 0$ there are no non-deterministic circuits C of size $S < 2^{n^\epsilon}$ such that

$$\mathbf{P}_y[C(y) = 1] - \mathbf{P}_x[C(g_n(x)) = 1] \geq 1/S$$

where y is taken at random from $\{0,1\}^{n+1}$ and x is random from $\{0,1\}^n$.

The order of the probabilities is important. While a nondeterministic circuit can simply guess the right seed x and verify its consistency with the observed string, it is now also forced to prove with a significant success if y is outside of the range of g_n.

It can be shown that the existence of super-bits is stronger than NP≠coNP but it is still considered by Rudich as a plausible assumption.

The attempt to prove the unprovability of P≠ NP in strong mathematical theories can be seen as an approach to extend the natural proofs barrier. Although, it is incorrect to say that as, hypothetically, any proof (of whatsoever) can be P/poly natural against P/poly. Nevertheless, the best known way how to employ natural proofs in the context of proof complexity works only for systems with the so called feasible interpolation.

3.2 Feasible interpolation

Feasible interpolation is a technique from proof complexity proposed by Krajíček [15]. It allows to extrapolate efficient circuits for certain problem using just existence of short propositional proofs.

Definition 8. *A proof system P admits feasible interpolation (EIP) iff there is a polynomial $p(x)$ such that for any disjunction $A(x,y) \vee B(x,z)$ (where x are the only common variables of A and B) with P-proof of size m there is a $p(m)$-size circuit $C(x)$ that for each assignment a to x outputs a tautology from the set $\{A(a,y), B(a,z)\}$.*

Proof systems like Resolution and Cutting Planes admit EIP, as proved by Krajíček [15] and by Pudlák [26] respectively.

Feasible interpolation can be employed to transform natural proofs barrier to a different kind of unprovability result, the hardness of tt-generators.

Theorem 3. *If there exists a SPRNG, then $tt_{s,k}$, where $s = k^{\omega(1)}$, is hard for any proof systems that admits feasible interpolation.*

This theorem comes from Krajíček [20] who mentioned that it uses a trick invented by Razborov [29].

Using stronger assumptions Rudich [33] extended the hardness of circuit lower bounds to potentially stronger systems that admit just a disjoint interpolation property [1], DIP: a proof system P admits DIP iff there exists a proof system R which have short proof of $A(x)$ or short proof of $B(y)$ whenever P proves efficiently $A(x) \vee B(y)$ (where x and y are disjoint tuples of variables).

[1] existential interpolation property in [33]

Theorem 4 (Rudich [33]). *If super-bits exist then $tt_{s,k}$, where $s = k^{\omega(1)}$, is hard for any proof system with DIP.*

Actually, Rudich does not use the whole power of DIP (see Theorem 29.2.3 in [20]). To prove the hardness of $tt_{3s,k}$, it is sufficient to assume just a circuit lower bound interpolation, CLB(s)-interpolation: a proof system P admits CLB(s)-interpolation iff for any $\tau(tt_{s,k})_f \vee \tau(tt_{s,k})_g$ with poly-size proof in P there is a poly-size P-proof of $\tau(tt_{t,k})_f$ or a poly-size P-proof of $\tau(tt_{t,k})_g$ for some $t = k^{\omega(1)}$.

If for any $f \notin$P/poly there is $t = k^{\omega(1)}$ such that proof system P proves $\tau(tt_{t,k})_f$ efficiently, then the system trivially admits CLB(s)-interpolation for $s \geq k^{\omega(1)}$. In fact, it implies the existence of an NP-natural property against P/poly. This observation yields the following fact.

Theorem 5. *If super-bits exist then for any proof system P there exists $f \notin P/poly$ such that $\tau(tt_{s,k})_f$, where $s = k^{\omega(1)}$, is hard for P.*

In other words, assuming the existence of super-bits, for any proof system there is a function $f \notin$P/poly such that proving superpolynomial lower bounds for f is hard for the proof system. It would be even more interesting if we obtained a more specific f, e.g. $f = SAT$. We will see in section 3.4 that the hardness of SAT$\notin$P/poly for all proof systems would imply NP$\subseteq$P/poly.

Feasible interpolation is nice, one could say canonical, example of extraction of a computation from short proofs. However, EF does not admit EIP unless the RSA cryptosystem is not secure [21]. And it is plausible that obtaining small interpolant from short proofs even in as weak systems as Res(2) (a natural extension of resolution working with 2-DNF formulas) would produce efficient algorithms for problems considered to be hard.

Therefore, it seems that one will need to investigate more subtle modifications of feasible interpolation if he wants to use it for systems like EF. For example, it is still open whether EF admits DIP (even under reasonable assumptions).

3.3 KPT theorem

Another technique from proof complexity, the so called KPT theorem takes advantage of the power of the completeness theorem from first-order logic and produces a certain kind of useful computations from first-order proofs.

In the last chapter of the thesis we will see that it can be used to obtain the unprovability of superpolynomial circuit lower bounds in the theory T_{NC^1}.

The KPT theorem was proven by Krajíček, Pudlák and Takeuti in [23].

Theorem 6 (KPT). *Let T be a universal theory over a language L which contains at least one constant or function symbol. Let $\phi(x, y, z)$ be an open L-formula and suppose T proves $\forall x \exists y \forall z \phi(x, y, z)$. Then there exists a finite sequence $t_1(x), t_2(x, z_1), ..., t_k(x, z_1, ..., z_{k-1})$ of L-terms (containing only the displayed variables) such that*

$$T \vdash \forall x \forall z_1, .., z_k \ \phi(x, t_1(x), z_1) \vee \phi(x, t_2(x, z_1), z_2) \vee ... \vee \phi(x, t_k(x, z_1, ..., z_{k-1}), z_k)$$

In particular, if L consists of functions that are efficiently computable then assuming provability of $\forall x \exists y \forall z \phi(x, y, z)$ the KPT theorem gives us efficiently computable functions $t_1, ..., t_k$ that can find in a certain sense the witness for the existence of y.

More precisely, on the input x, $t_1(x)$ produces a witness for y (such that for all z_1's $\phi(x, t_1(x), z_1)$ is true) or t_1 makes a mistake and there exists z_1 such that $\neg\phi(x, t_1(x), z_1)$ but then having this z_1 t_2 produces a potential witness for y etc. After finitely many steps one of the functions $t_1, ..., t_k$ will certainly succeed.

This can be seen as an interactive protocol and it is indeed very useful as we will see in the last chapter.

3.4 Formulas hard for all proof systems

We mentioned that the existence of super-bits implies that for any proof system there is a function $f \notin$P/poly such that proving $f \notin$P/poly is hard for P.

By definition of propositional proof systems the reasonable computational assumption NP$\neq$coNP itself implies the existence of hard tautologies (for any proof system). In fact, it implies the existence of a sequence of tautologies hard for all propositional proof systems.

To see this let P_i be the i-th propositional proof system in the enumeration of all propositional proof systems and $t_{i,j}$ be a sequence of hard tautologies for P_i which exists according the assumption. Then tautologies $t_i = t_{1,i} \wedge ... \wedge t_{i,i}$ must be hard for all proof systems.

This can be seen also as a generalization of the trivial observation: if the hardest tautologies for a specific system P are easy then P is p-bounded.

As t_i's encode in a sense all tautologies they are obviously the hardest ones. We, however, need a more specific example of hard tautologies, circuit lower bounds.

It is indeed an emerging question whether we can somehow transform arbitrary hard tautologies into hard circuit lower bounds and next sections present some techniques with a partial success in this direction.

However, if superpolynomial circuit lower bounds are really hard for all proof systems, then NP$\subseteq$P/poly:

If SAT is not in P/poly, then for some $s = k^{\omega(1)}$ formulas $\tau(tt_{s,k})_{\chi_{SAT_k}}$ where χ_{SAT_k} is the characteristic function of SAT restricted to inputs of length k, have short proofs in the following proof system: the proof system upon receiving a formula $\tau(tt_{s,k})_b$ where b is a string of length 2^k checks whether b is or is not equal to χ_{SAT_k}. If so, it accepts $\tau(tt_{s,k})_b$ as a tautology, otherwise it proceeds as, say, EF. Deciding this property of b can be done in $poly(2^k)$-time, hence this system is a proof system in the sense of Cook and Reckhow.

Therefore, proof of the hardness of NP$\not\subseteq$P/poly for all proof systems, resp. proof of this specific kind of its unprovability, already implies the existence of efficient circuits for hard problems.

On the other hand, using derandomization arguments it is possible to show that NEXP$\subseteq$P/poly implies hardness of superpolynomial circuit lower bounds for

all proof systems as proved in [20]. Similarly, assuming BPP$\not\subseteq$NP, this is the case for exponential circuit lower bounds as independently observed by Impagliazzo (see a footnote in [17] Section 1) and by Alekhnovich (see [31] Section 1.1). Of course, these arguments use unreasonable assumptions (for our purpose) but one could think how to make them weaker e.g. assume that BPP contains problems that are hard for constant-depth Frege and obtain the hardness of τ-formulas just for constant-depth Frege systems.

3.5 Iterability

While hardness of superpolynomial circuit lower bounds for all proof systems would have strong consequences the task of converting arbitrary hard formulas into hard circuit lower bounds is not hopeless. Krajíček already invented a technique of iterability which allows him to prove that *tt*-generators are in a sense the hardest generators, i.e. it allows him to transform arbitrary generator that is hard in a certain way into hard *tt*-generators.

Definition 9. *Function $g : \{0,1\}^n \mapsto \{0,1\}^m$ is (exponentially) iterable for proof system P iff no disjunction of the form*

$$\tau(g_n)_{b_i}(x^1) \vee ... \vee \tau(g_n)_{b_{t(n)}}(x^{t(n)})$$

has P-proof of polynomial (resp. $2^{n^{\Omega(1)}}$) size in n. Here x^i's in the notation $\tau(g_n)_{b_i}(x^i)$ are disjoint n-tuples of atoms standing for inputs of g_n in corresponding $\tau(g_n)_{b_i}$ formulas, $t(n)$ is arbitrary function, and $b_1, ..., b_{t(n)}$ are m-tuples of variables and constants such that:

- $b_1 \in \{0,1\}^m$
- *variables occurring in b_i are among $x^1, ..., x^{i-1}$, for $i \leq t(n)$*

Note that (exponential) iterability implies hardness for $m = poly(n)$ (resp.$m \leq 2^{O(n^{\Omega(1)})}$) as one can take $t(n) = 1$.

Theorem 7 (Krajíček[17])**.** *Assume that proof system P simulates resolution and that there exists a g exponentially-iterable for P. Then, there is $c > 0$ such that the truth table function $tt_{n^c,n}$ is hard for P.*

Now, if we want to prove the hardness of superpolynomial circuit lower bounds it is sufficient to find an exponentially-iterable generator.

3.6 Gadget generators

As a next partial result in the attempt to transform arbitrary hard tautologies into hard (or even iterable) generators we present another construction by Krajíček [18]. It produces a generator hard for any proof system which cannot prove PHP efficiently.

Definition 10. *Let $k, t \geq 1$ be any parameters such that $t > k(k+1)$. Put $n = k(k+1+t)$ and $m = (k+1)t$. Hence $m > n$.*

Map $g_{k,t} : \{0,1\}^n \mapsto \{0,1\}^m$ *is defined as follows. Input string* x *of length* n *interpret as*

$$x = (v, u^1, ..., u^t)$$

where $v = (v_{i,j})_{i\in[k+1],j\in[k]}$ *and* $u^s = (u^s_j)_{j\in[k]}$ *for* $s = 1, ..., t$.

The output string y *of length* m *is defined as* $y = (y^1, ..., y^t)$ *where*

$$y^s_i = \bigvee_{j\in[k]} (v_{i,j} \wedge u^s_j)$$

for $s = 1, ..., t$.

This construction can be generalized so that y^s is defined by a more complicated function. The string v is called a gadget. In our case v is supposed to be interpreted as a graph of a function between $[k]$ and $[k+1]$. Then it yields the following.

Theorem 8 (Krajíček [18])**.** *Let* $d \geq 2, k \geq 1$ *and* $t = k^2 + k + 1$. *Then, with* $n = k(k + 1 + t)$ *as above, the map*

$$g_{k,t} : \{0,1\}^n \to \{0,1\}^{n+1}$$

is an exponentially hard proof complexity generator for constant depth Frege. In fact, it is hard generator even for all stronger systems for which PHP is hard.

It seems doable to improve this result to iterability of the generator $g_{k,t}$. Consequently, this would imply the hardness of superpolynomial circuit lower bounds for all systems that cannot prove PHP efficiently. In particular, for constant depth Frege.

As producing hard generators for a system P just from hard formulas might be hard it could be helpful to assume in addition e.g. the existence of a stronger system Q. Then P does not prove soundness of Q efficiently, cf [14]. Furthermore, we could assume that there are no hard generators for Q etc. Unfortunately, we do not see now how to use those properties even in the case of constant depth Frege (=P) which is known to be weaker than Frege (=Q).

On the other hand, it is possible to employ a hardness assumption in the following tricky way. Jeřábek [12] introduced a system WF extending EF by a rule simulating the dual weak PHP. Although WF is a quite natural system one can say that in a sense it captures the power of generators. Then, if EF does not simulate WF, generators are hard for EF. Details can be found in [17].

3.7 The best known hardness result for circuit lower bounds

We have presented few techniques converting various computational assumptions to the hardness of superpolynomial circuit lower bounds. However, one can say that the best known hardness results concerning circuit lower bounds in proof complexity are unconditional.

Raz [27] and Razborov [30] proved the hardness of circuit lower bounds for resolution. They were, however, using a different formalization in which the formulas are easy even for constant-depth Frege systems.

Razborov [31] later obtained also unconditional hardness of $tt_{n^{\omega(1)},n}$ even for systems that are not known to admit feasible interpolation, $Res(\epsilon log n)$, a resolution working with $\epsilon log n$-DNF formulas (instead of clauses). Here $\epsilon > 0$ is a sufficiently small constant.

His result is based on proving the hardness of NW-generators. We will not go into details of his proof but we dedicate the rest of the thesis to investigation of the NW-generators. We will see that they in fact give us another elegant way how to employ hardness assumptions to produce hard tautologies.

4 Nisan-Wigderson generators

As Razborov pointed out in [31], to prove the hardness of $\tau(tt_{t(n),n})_f$ in a proof system, it is sufficient to show that there exists a generator $g : \{0,1\}^{t_0(n)} \longmapsto \{0,1\}^{2^n}$ with arbitrary $t_0(n) \leq 2^{O(n)}$ and such that g is

1. constructive: for every $x \in \{0,1\}^{t_0(n)}$, there is a $t(n)$-size circuit computing y-th bit of $g(x)$ from $y \in \{0,1\}^n$

2. hard: it is hard to prove $f \notin Rng(g)$ in the given proof system

Condition 1. means that for each $x \in \{0,1\}^{t_0(n)}$, the function given by the truth table $g(x)$ is computable by $t(n)$-size circuits. Therefore, since by 2. it is hard to prove that f differs from all $g(x)$, it is also hard to prove that it is not computable by a $t(n)$-size circuit.

A promising example of a constructive generator in the above sense is inspired by the Nisan-Wigderson generators (shortly NW-generators), cf. [24].

Definition 11. *Let $n < m$ and A be an $m \times n$ 0-1 matrix with l ones per row. $J_i(A) := \{j \in [n] = \{1, ..., n\} | A_{ij} = 1\}$. Let $f : \{0,1\}^l \longmapsto \{0,1\}$ be a Boolean function. Define function $NW_{A,f} : \{0,1\}^n \longmapsto \{0,1\}^m$ as follows: The i-th bit of the output is computed by f from the bits $x|J_i(A)$, these are x_k's such that $k \in J_i(A)$.*

We speak about these functions as about NW-generators but in computational complexity the term NW-generator usually refers to the construction where f is a suitably hard function and A is in addition a (d, l) combinatorial design. The design property means that $J_i(A) \cap J_k(A)$ has size $\leq d$ for any two different rows i, j.

Assuming that the NW-generators are based on the combinatorial designs with the same parameters as in the seminal paper [24], Razborov proposed,

Conjecture 1 (Razborov [31])**.** *Any NW-generator based on any poly-time function that is hard on average for $NC^1/poly$, is hard for the Frege proof system.*

Conjecture 2 (Razborov [31])**.** *Any NW-generator based on any function in $NP \cap coNP$ that is hard on average for $P/poly$, is hard for Extended Frege.*

The parameters are actually not specified more precisely in [31]. In the main construction of design matrices in Lemma 2.5 in [24] we have $d = \log m \leq l$ and $n = O(l^2)$.

It is also not clear what exactly it means to be hard on average but it is not important. The idea is obvious. We want to use functions that are hard maybe even in some weaker sense. In the next section we will see an example of such hardness.

Note that if $\log m = \sqrt{n}$ it is like to have a generator sending strings of length n^2 to strings of length 2^n. Thus in the view of the above discussion if there are poly(n)-size circuits computing the set $J_i(A)$ from given i (and the designs constructed in Lemma 2.5 in [24] have this property), then the NW-generators based on functions computed by poly(n)-size circuits are constructive

(for a polynomial $t(n)$). Therefore, the hardness of such NW-generators implies the hardness of superpolynomial circuit lower bounds.

Intuitively, there are two main properties that could make NW-generators hard: locality and pseudorandomness.

Locality: If the base matrix A is a combinatorial design then the prover attempting to prove $b \notin Rng(NW_{A,f})$ has to show that the system of equations $f(x(J_i)) = b_i, 1 \leq i \leq m$ has no solution. But as the tuples of equations have small number of common variables the system might look consistent. In terms of Buss-Pudlák games, cf [6], this could allow Sam to cheat Pavel. In section 4.3 we show a situation where this can be indeed achieved. Also already known results use designs, resp. expanders (see next section).

Pseudorandomness: NW-generators based on suitably hard functions are good pseudorandom generators in the sense that certain computational models can hardly distinguish theirs outputs from the random ones. If the range of the generator looks random it might be also hard to determine (for certain proof systems) whether the given string is outside of the range. Now, note that a non-deterministic guessing of the right seed can easily determine elements inside the range but it does not seem so easy to determine elements that are outside. This is in fact expressed in the Rudich's super-bit conjecture (the existence of super-bits).

4.1 Previous results

In [2] Alekhnovich et al. proved that the NW-generators are hard for resolution in terms of width (size of the biggest clause in the proof).

Theorem 9 (Alekhnovich et al. [2] (oversimplified)). *Let A be an $m \times n$ (d,l)-design, f be PARITY function on l variables (i.e. it is 1 on input x if x contains even number of ones), and $b \in \{0,1\}^m$. Then every resolution proof of $\tau(NW_{A,f})_b$ (encoded in a specific way) must have width $> \frac{r(l-rd)}{2l}$ (for any r).*

Typically, we have parameters like $l = \sqrt{n}, r = d = n^{1/6}$, so the width is $> n^{1/6}/2 - 1/2$. By Ben-Sasson and Wigderson [3] this says that there are no subexponential-size proofs of $\tau(NW_{A,f})_b$ in tree-like resolution.

Theirs result gives also the hardness for resolution (and as they show even for the system PCR, a natural extension of Polynomial Calculus and resolution) but only for the NW-generators stretching n bits to $\leq o(n^2)$ bits. This input/output ratio hindered to obtain hardness of superpolynomial circuit lower bounds for resolution.

However, as we already mentioned, Razborov [31] established the hardness of the NW-generators even for certain $Res(k)$ systems, resolution working with k-DNF formulas. He obtained various forms of the following statement.

Theorem 10 (Razborov [31]). *Let A be an $m \times n$ (r,d)-lossless expander, and assume that*

$$min_{i \in [m]} |J_i(A)| \geq Cd(k + \log m)$$

for a sufficiently large constant $C > 0$. Let $\leq$ be an arbitrary ordering of A and $b \in \{0,1\}^m$. Then, every $Res(k)$ refutation of $\tau_{\leq}(A,b)$ must be of size $\geq exp(r/2^{O(kd)})$.

Again, if we oversimplify it and apply the right parameters, this says that for $f = PARITY$ and A a suitable expander (a version of design matrix the existence of which he also proved) formulas $\tau(NW_{A,f})_b$ encoded in a specific way are hard for $Res(k)$ where k depends on the choice of parameters. Razborov then shows that this gives the hardness of superpolynomial circuit lower bounds for $Res(\epsilon \log n)$ where $\epsilon > 0$ is sufficiently small.

We will not describe this result in more details. Instead we will show how to obtain the hardness of the NW-generators with big input/output ratio for systems with forms of feasible interpolation. In particular, for resolution or tree-like resolution. This result will not use design properties of the base matrix but a hardness of base functions.

Then, in section 4.3 we will show the hardness of the NW-generators with big input/output ratio for systems roughly described as tree-like Frege systems working with formulas using only a small fraction of variables of the given NW-generator. Here we use locality properties captured in design matrices.

The big input/output ratio allow us to interpret these lower bounds as the hardness of superpolynomial circuit lower bounds.

4.2 NW-generators in proof systems with EIP

We will now show that the NW-generators based on certain computationally hard functions are hard for proof systems admitting feasible interpolation [25].

There is a simple idea illustrating this result. Consider tautology $f(x) \neq 0 \vee f(x) \neq 1$ where f is a boolean function. Assume that it can be expressed by a poly-size formula (e.g. $f \in NP \cap coNP$ is a sufficient condition for this). If there was a poly-size proof of this tautology in a proof system with feasible interpolation, there would exist a poly-size circuit C that could decide for every assignment a to x whether $f(a) \neq 0$ or $f(a) \neq 1$, hence it would compute function f. This means that if $f \in NP \cap coNP$ is not in P/poly, tautology $f(x) \neq 0 \vee f(x) \neq 1$ is hard for any proof system with EIP.

This idea can be extended into the context of NW-generators. We show how to do it, firstly, in the case there is an additional condition on the base matrix A, uniformity.

Definition 12. *Let A be an $m \times n$ 0-1 matrix with l ones per row. $J_i(A) = \{j \in [n] | A_{i,j} = 1\}$. A is l-uniform iff there is a partition of $[n]$ into l sets such that there is exactly one element of each $J_i(A)$ in each set of the partition.*

Note that $m \times n$ $(\log m, l)$ design matrices with $l = \sqrt{n}$ ones per row constructed in the proof of Lemma 2.5 in [24] are $\sqrt{n}$-uniform.

Theorem 11. *Any NW-generator based on*

1. *any $m \times n$ l-uniform matrix A with l ones per row*
2. *any function $f : \{0,1\}^l \longmapsto \{0,1\}$ in $NTime(m^{O(1)}) \cap coNTime(m^{O(1)})$ such that f does not have $m^{O(1)}$-size circuits*

is hard for any proof system P with EIP.

Before we give the proof let us say that the assumption $f \in NTime(m^{O(1)}) \cap coNTime(m^{O(1)})$ (which already appeared in a stronger form in the second Razborov's Conjecture) allows us to express formula $\tau(NW_{A,f})_{(b_1,\ldots,b_m)}$ as $m^{O(1)}$-size formula

$$\bigvee_{i \leq m} \neg\alpha_{b_i}(x|J_i(A), v^i)$$

using $NTime(m^{O(1)})$-definitions of $f(x|J_i(A)) = \epsilon$, for $\epsilon = 0, 1$:

$$f(x|J_i(A)) = \epsilon \text{ iff } \exists v\ (|v| \leq m^{O(1)}\ \alpha_\epsilon(x|J_i(A), v)$$

where α_ϵ is a polynomial time relation. The tuples of variables v^i in the disjunction are disjoint.

Proof. Assume that there is a proof system P with EIP and $m^{O(1)}$-size proof of some $\tau(NW_{A,f})_b$. This $\tau(NW_{A,f})_b$ can be expressed in a form

$$\bigvee_i \neg\alpha_0(x|J_i(A), v^i) \vee \bigvee_j \neg\alpha_1(x|J_j(A), v^j)$$

where $\neg\alpha_0(x|J_i(A), v^i)$ encodes $f(x|J_i(A)) \neq 0$ and $\neg\alpha_1(x|J_j(A), v^j)$ encodes $f(x|J_j(A)) \neq 1$.

By EIP, there exists an $m^{O(1)}$-size circuit C that for every assignment a to x finds out which of $\bigvee_i \neg\alpha_0(a|J_i(A), v^i)$, $\bigvee_j \neg\alpha_1(a|J_j(A), v^j)$ is true.

Denote now by S a partition of $[n]$ certifying that A is l-uniform. Define a linear order on S by the smallest elements of its blocks: $K < L$ for $K, L \in S$ iff $minK < minL$. An $m^{O(1)}$-size circuit computing f proceed as follows.

It extends input $a \in \{0,1\}^l$ to $\overline{a} \in \{0,1\}^n$ where $\overline{a}_i$ for $i \in K_j$, K_j the j-th smallest block of S, has the same value as a_j. Then it uses the circuit C to find out which of $\bigvee_i \neg\alpha_0(\overline{a}|J_i(A), v^i)$, $\bigvee_j \neg\alpha_1(\overline{a}|J_j(A), v^j)$ is true. If it is the former one, then it outputs 1, otherwise 0.

This circuit finds the true value of $f(a)$ because the uniformity of A implies that if $\bigvee_i \neg\alpha_0(\overline{a}|J_i(A), v^i)$ then all $\neg\alpha_0(\overline{a}|J_i(A), v^i)$'s hold, resp.
if $\bigvee_j \neg\alpha_1(\overline{a}|J_j(A), v^j)$ then all $\neg\alpha_1(\overline{a}|J_j(A), v^j)$'s hold. □

In order to obtain the unprovability of superpolynomial circuit lower bounds we need to scale down the previous result.

Define a formula interpolation property, FIP, as EIP where the resulting circuit C is in fact a formula. For the reader who is familiar with the proof of feasible interpolation for resolution it is easy to see that tree-like resolution admits FIP. Here, tree-likeness means that each intermediate formula in a proof can be used at most once in subsequent derivations.

Theorem 12. *Any NW-generator based on any $m \times n$ l-uniform matrix A with l ones per row, and on any poly-time function in l which does not have poly-size formulas in m is hard for any proof system P with FIP.*

Proof. If we replace EIP by FIP in proof of Theorem 11, we obtain a poly-size formula computing f. □

Theorem 12 implies a conditional hardness of superpolynomial circuit lower bounds.

Firstly, note that it is easy to construct an $m \times n$ l-uniform matrix for $m = 2^{n^\delta}$, where $\delta < 1$. The resulting NW-generator based on poly-time functions are constructive in the following sense: for any $x \in \{0,1\}^{n^{1/\delta}}$ the function represented by the truth table $NW(x)$ is computable by poly-size circuits in n. Therefore, Theorem 12 implies that if there exists a poly-time function hard for subexponential size formulas, then it is hard to prove any superpolynomial circuit lower bound (i.e. formulas $\tau(tt_{t(n),n})_f$ for any superpolynomial function $t(n)$ and for any function f) in proof systems with FIP.

This applies e.g. to tree-like resolution.

Next we show how to obtain the hardness of the NW-generators even without the uniformity assumption. We, however, need to use a stronger form of interpolation, CIP.

Definition 13. *A proof system P admits constructive interpolation property (CIP) iff there is a family of polynomial size circuits $\{C_n\}_{n=1}^\infty$ such that for any disjunction $A(x,y) \vee B(x,z)$ (where x are the only common variables of A and B) with P-proof π of size m there is a circuit $C(x,\pi) \in \{C_n\}_{n=1}^\infty$ that for each assignment a to x outputs an $O(m)$-size proof for a tautology in $\{A(a,y), B(a,z)\}$. Note that the input of the circuit C contains π, so it has polynomial size size in the length of π.*

Krajíček's [15] proof that resolution admits EIP actually gives also CIP. Pudlák [26] later gave a different proof of CIP with better bound on proofs: the constructed proof is of size $\leq m$.

Theorem 13. *Any NW-generator based on*

1. *any $m \times n$ 0-1 matrix A with l ones per row (not necessarily a combinatorial design)*
2. *any function $f : \{0,1\}^l \longmapsto \{0,1\}$ in $NTime(m^{O(1)}) \cap coNTime(m^{O(1)})$ such that for any $m^{O(1)}$-size circuit C, $|\mathbf{P}_{x \in \{0,1\}^l}[C(x) = f(x)] - \frac{1}{2}| < \frac{1}{2m}$*

is hard for any proof system P with CIP.

Proof. Assume that there is a proof system P with CIP and $s = m^{O(1)}$-size P-proof of some $\tau(NW_{A,f})_{(b_1,\ldots,b_m)}$. We will describe an $m^{O(1)}$-size circuit C such that $|\mathbf{P}_{x \in \{0,1\}^l}[C(x) = f(x)] - \frac{1}{2}| \geq \frac{1}{2m}$.

Our f is in $NTime(m^{O(1)}) \cap coNTime(m^{O(1)})$. As we noted, this means that $\tau(NW_{A,f})_{(b_1,\ldots,b_m)}$ can be expressed as

$$\bigvee_{i \leq m} \neg\alpha_{b_i}(x | J_i(A), v^i)$$

CIP implies that there is an $m^{O(1)}$-size circuit which for any assignment a to the variables x outputs proof of one of the disjunctions

$$\bigvee_{i=1}^{k} \neg\alpha_{b_i}(a | J_i(A), v^i), \quad \bigvee_{i=k+1}^{m} \neg\alpha_{b_i}(a | J_i(A), v^i)$$

where $k = \lfloor \frac{m}{2} \rfloor$. The new proof has the size at most $O(s)$. Therefore, we can iterate the usage of CIP $\log m$ times and get the true value of some $f(a|J_i(A))$. The resulting circuit C' consisting of all circuits given by CIP remains $m^{O(1)}$-size and for any input a it outputs the true value of some $f(a|J_i(A))$.

Fix an $i \in [m]$ such that C' outputs the value of $f(a|J_i(A))$ for at least $\frac{2^n}{m}$ $a's \in \{0,1\}^n$. Now, let C be an $m^{O(1)}$-size circuit which uses C' to check whether given input leads to the fixed value of $f(a|J_i(A))$. If it does, then it outputs the value of $f(a|J_i(A))$, otherwise it outputs always zero or always one, whichever is better on the remaining inputs. Therefore,

$$\mathbf{P}_{x\in\{0,1\}^n}[C(x) = f(x|J_i(A))] \geq \frac{1-1/m}{2} + \frac{1}{m} = \frac{1}{2} + \frac{1}{2m}$$

Since $f(x|J_i(A))$ does not depend on all bits of $x = x_1, ..., x_n$ we can rewrite $\mathbf{P}_{x\in\{0,1\}^n}[C(x) = f(x|J_i(A))]$ as the average over all possible choices of values of bits from $[n] \setminus J_i(A)$ of the same expression where only $x|J_i(A)$ are chosen at random. It follows that for some particular choice of these additional values the circuit C preserves the advantage. □

Note that we could obtain analogous results if we weakened EIP to allow bigger size of the resulting circuit. Then we would need to assume just the existence of computationally harder base functions (whose hardness would depend on the size of the circuits allowed by such a modification of EIP) for the NW-generators.

4.3 NW-generators in strongly-sound local systems

The previous result is based on the hardness of base functions. Here we show another one which relies on the combinatorial designs of NW-generators. In fact, it could be presented as a strategy for Sam to cheat Pavel in certain Buss-Pudlák games but we prefer more straightforward presentation of our argument suggested by Krajíček.

The hardness result holds for proof systems that actually do not have to be proof systems in Cook-Reckhow sense. This is because theirs proofs do not have to be efficiently verifiable. However, these systems are restricted in other ways. Mainly, they work with formulas that depend somehow only on a small fraction of the base matrix of the given NW-generator.

Definition 14. *Consider an NW-generator based on a matrix A and a function f. A formula C is s-local if it is disjunction of formulas (not necessarily literals) F_i such that for each F_i there are some formulas $f(x|J_{j_k}(A)) = b_{j_k}$ $k = 1, ..., s$ containing every variable of F_i that occurs in some $f(x|J_j(A)) = b_j$.*

Definition 15. *A derivation rule allowing to derive an s-local formula F from s-local formulas G, H is strongly-sound if it is sound and whenever a partial assignment of variables of G, H satisfies some disjunct in G and some disjunct in H, it satisfies also some disjunct in F.*

For example, the resolution rule deriving $C \vee B$ from $C \vee x$ and $B \vee \neg x$ is strongly sound. Whenever you have a partial assignment satisfying some literal in $C \vee x$ and some literal in $B \vee \neg x$ it has to satisfy also some literal in $B \vee C$.

On the other hand, if a rule is sound it does not have to be strongly-sound. A sound rule might derive a tautology A from formulas B and C while no partial assignment of variables of B and C satisfies a disjunct in A.

Definition 16. *An $SL(s)$-proof of $\tau(NW_{A,f})_b$ is any derivation of empty clause from clauses representing $\neg\tau(NW_{A,f})_b$ consisting of s-local formulas and using strongly-sound derivation rules.*

Say that $SL(s)$ proof of any other tautology is a Frege proof. Then, in particular, we can see $SL(s)$-proofs as Frege proofs restricted only in proving $\tau(NW_{A,f})_b$ formulas to work only with disjunctions of formulas depending on at most s lines of given generator.

We prove the hardness of the NW-generators for tree-like version of this $SL(s)$ system. More precisely, if the NW-generator is based on an $m \times n$ matrix then we have $s = n^{1/2-\epsilon}$. This system is probably not very strong. It is almost like tree-like $(1/2 - \epsilon)logn$-depth Frege working with formulas with connectives of bounded arity. But it can be shown that short tree-like proofs in such Frege system can be simulated by small width Resolution proofs and consequently by short Resolution proofs as Razborov pointed out to me.

We say that a boolean function f is robust if any partial assignment that does not assign all input variables of f has two extensions a and b such that $f(a) = 1$ and $f(b) = 0$. An example of a robust functions is PARITY.

Theorem 14. *Any NW-generator based on any $m \times n$ $(\log m, \sqrt{n})$ combinatorial design and on any robust function f is hard for tree-like $SL(n^{1/2-\epsilon})$. In fact, it requires $SL(n^{1/2-\epsilon})$-proofs of size $> (3/2)^{n^\epsilon}$. Here $\epsilon > 0$ is arbitrary and $m < 2^{n^{\epsilon/2}}$.*

Note that the result is unconditional as robust functions and $(\log m, \sqrt{n})$ designs exist.

Proof. For the sake of contradiction, assume there is a $(3/2)^{n^\epsilon}$-size tree-like $SL(n^{1/2-\epsilon})$-proof π_0 of $\tau(NW_{A,f})_b$ for some b.

By Spira Lemma we can pick a formula C_1 in the proof such that the size of tree-like derivation of C, a subproof π_1 of π_0, is between $1/3|\pi_0|$ and $2/3|\pi_0|$.

If there is a formula F_i in $C = \bigvee_i F_i$ which can be satisfied by solving some $\leq n^{1/2-\epsilon}$ equations $f(x|J_i(A)) = b_i$, denote the partial assignment of all variables in $\tau(NW_{A,f})_b$ by a_1 and go to $\pi_0 - \pi_1$ (the proof π_0 without the derivation π_1). Otherwise, let a_1 be empty and go to π_1.

In i-the step, we are in a subproof π_{i-1} of π_0 and have an assignment a_{i-1} solving some $\leq (i-1)n^{1/2-\epsilon}$ lines of the generator. In π_{i-1} we pick again a formula C_i such that the size of tree-like derivation of C_i, a subproof π_i of π_{i-1}, is between $1/3|\pi_{i-1}|$ and $2/3|\pi_{i-1}|$. If there is an extension a_i of a_{i-1} solving some $\leq in^{1/2-\epsilon}$ lines of the generator and satisfying some disjunct of C_i we go to $\pi_{i-1} - \pi_i$. Otherwise, proceed to π_i.

Design property and robustness guarantee existence of an assignment solving any $n^{1/2-\epsilon}$ new lines of the generator even after $n^{\epsilon/2}-1$ iterations: any assignment

c solving some $k < n^{1/2-\epsilon/2}$ equations $f(x|J_i(A)) = b_i$ assigns at most $k \log m < n^{1/2}$ variables from $x|J_j(A)$ for arbitrary new equation $f(x|J_j(A)) = b_j$. Thus, there is a free variable in $x|J_j(A)$ which by robustness of f can be set so that the resulting assignment extending c solves also $f(x|J_j(A)) = b_j$.

Each axiom is a clause from representation of an equation $f(x|J_i(A)) = b_i$ so it can be satisfied in any step $i \leq n^{\epsilon/2}$ of our construction. On the other hand, empty clause cannot be satisfied.

Therefore, as the proof is short, after $n^{\epsilon/2}$ iterations we will find a triple A, B, C of $n^{1/2-\epsilon}$-local formulas where C is derived by a strongly-sound rule from A and B such that a partial assignment a satisfies a disjunct in A and a disjunct in B while no F_i from $C = \bigvee_i F_i$ is satisfied by an extension of some subassignment of a. This is contradiction because by strongly-sound rule a satisfies a formula F_i in C. □

As m can be as large as $2^{n^{\Omega(1)}}$ and the base function (PARITY) is in P/poly we obtain the hardness of superpolynomial circuit lower bounds for $SL(n^{1/2-\epsilon})$.

5 NW-generators in strong proof systems

Ajtai's method of proving lower bounds requires to show a certain kind of unprovability of a universal statement $(\forall z Sat(n, z, T_n))$. Naturally, it might be easier to prove the unprovability of a statement with the higher quantifier complexity. Indeed, Krajíček found out that in the case of NW-generators it can be done.

His technique applies for the theory T_{PV} which is the true universal theory of $\mathbf{N}$ in the language L_{PV} that consists of all poly-time functions and relations. The theory T_{PV} proves Ref_P for any other proof system. Thus, if one could modify it to work for universal statements it would prove that NW-generators based on hard functions are hard for all proof systems.

Definition 17. *Let $AH_f(l,k)$ (approximating hardness) be the minimal s such that there is a size s circuit C with l inputs such that*

$$P_{u\in\{0,1\}^l}[C(u) = f(u)] \geq 1/2 + l^{-k}$$

Theorem 15 (Krajíček [19])**.** *Let A_n be an $(n+1)\times n$ $(\log(n+1), l)$ combinatorial design. Assume that f is an $NTime(n^{O(1)}) \cap coNTime(n^{O(1)})$ function on l variables with unique witnesses. Further, let M be a non-standard model of true arithmetic, $n \in M$ its non-standard element, and let $b \in M$ be any string of length $n+1$ that is outside of the range of $NW_{A_n,f}$.*

If, for all fixed $k \geq 1$, $AH_f(l,k)$ is a super-polynomial function of l then there exists a model N of T_{PV} that is a cofinal extension of M_n and a string $w \in N$ of length n such that in N it holds:

$$\forall i \in [n+1]\ f(w(J_i)) = b_i$$

The model N obtained above certifies the unprovability of the statement

$$\forall x(|x| = n)\exists i \in [n+1]\forall y(|y| \leq n^{O(1)})\neg A_{b_i}(x(J_i(A)), y) \qquad (*)$$

In order to obtain a super-polynomial lower bound on EF (or even all propositional proof systems) we need to derive a contradiction from the assumption that there are short proofs of $b \notin Rng(NW_{A,f})$, i.e. from the assumption that PV (resp. T_{PV}) proves

$$\forall x(|x| = n)\forall y(y = (y_1, ..., y_{n+1}))\exists i \in [n+1]\neg A_{b_i}(x(J_i(A)), y_i)$$

One could accomplish that by showing that the following collection schema holds in N:

$$\forall i \exists y_i B(y_i) \to \exists y = (y_1, ..., y_m)\forall i B(y_i)$$

where i is sharply bounded by m, y_i, y are bounded and B is an open L_{PV} formula.

Cook and Thapen [9], however, showed that provability of the collection scheme in T_{PV} would imply the existence of an efficient algorithm for factoring. We discuss this in more details in the section 5.2.

As we are interested in the unprovability of superpolynomial circuit lower bounds we want to reformulate Theorem 15 into the context of the first Razborov's conjecture, i.e. we want f to be P/poly instead of NP∩coNP function and m to be as large as $2^{n^{\Omega(1)}}$.

Denote by $T_{Fla(m)}$ the true universal theory of $\mathbf{N}$ in the language $L_{Fla(m)}$ that consists of all functions and relations computable by $poly(m)$-size formulas. (This is a natural generalization of T_{NC^1}.) Then the following statement is the reformulation we wanted. It can be interpreted as the unprovability of P$\neq$NP by proving circuit lower bounds in $T_{Fla(2^{n^{\Omega(1)}})}$.

Theorem 16. *Let A_n be an $m \times n$ $(\log m, l)$ combinatorial design where $n < m \leq 2^{n^\epsilon}$, $\epsilon < 1$. Assume that f is a function on l variables computable by poly(n)-size circuits. Further, let M be a non-standard model of true arithmetic, $n \in M$ its non-standard element, and let $b \in M$ be any string of length m that is outside of the range of $NW_{A_n,f}$.*

If there are no poly(m)-size formulas computing f on at least a fraction of $\frac{1}{2} + \frac{1}{m^{O(1)}}$ of all inputs then there exists a model N of $T_{Fla(m)}$ that is a cofinal extension of M_n and a string $w \in N$ of length n such that in N it holds:

$$\forall i \in [m]\ f(w(J_i)) = b_i$$

Krajíček's proof of Theorem 15 gives us proof of this Theorem as well. One just needs to observe that in his proof the provability of $(*)$ in T_{PV} yields P/poly circuits approximating f in the same way as the provability of (*) (properly expressed) in $T_{Fla(m)}$ yields $poly(m)$-formulas approximating f.

To express the statement $(*)$ properly in $T_{Fla(m)}$ (or in T_{NC^1}) we can use the observation that for any function $f \in$P/poly there are NC^1 predicates A_1 and A_0 such that for $a = 0, 1$

$$f(u) = a \text{ iff } \exists y(|y| \leq |u|^{O(1)} \wedge A_a(u, y)) \quad (1)$$

Here, the witnesses y might be e.g. computations of $P/poly$ circuits of f on inputs u.

Again, if we obtained the collection scheme in $T_{Fla(m)}$, the argument of Cook and Thapen [9] would give us an efficient algorithm for inverting $Fla(m)$ functions (in a certain sense), see the section 5.2.

5.1 Interactive communication and KPT theorem

The key observation in the proof of Theorem 16 is that non-existence of the desired model N together with the KPT theorem gives us a specific interactive communication solving certain computational task described in the following paragraphs.

The interactive communication model is interesting on its own. It can be shown that hard functions produce a computational task which is hard for this model. Thus, there is a potential for its further independent use.

Let A be an $m \times n$ 0-1 matrix with l ones per row where $n < m$ and f be a function on l variables computed by $poly(m)$-size circuits. Fix $b = (b_1, ..., b_m) \in \{0, 1\}^m$ any string outside of the range of $NW_{A,f}$. In this situation Krajíček [19] defines the following task.

Computational Task (T): *Given $x \in \{0,1\}^n$ find $i \in [m]$ such that i-th bit of $NW_{A,f}(x)$ differs from b_i.*

Then, there is a specific interactive model for solving (T) introduced in [23] as an interpretation of Herbrand theorem, and formalized in terms of computational classes in [22]. The model in which two players, a computationally limited Student and an unrestricted Teacher, interact proceeds as follows. In the first step:

- The Student, upon receiving an input $x \in \{0,1\}^n$, computes his first candidate solution $i_1 \in [m]$
- If i_1 solves (T) the computation stops
- If i_1 fails to solve (T) the Teacher sends to the Student a witness y_1 to $f(x(J_{i_1}(A)) = b_{i_1}$

In general, in the k-th step the Student computes a candidate solution $i_k \in [m]$ from x and from the witnesses $y_1, ..., y_{k-1}$ he has received from the Teacher in the previous $k-1$ steps. If i_k solves (T) the computation stops, if not the Teacher sends to the Student a witness y_k certifying the incorrectness, i.e. witnessing $f(x(J_{i_k}(A))) = b_{i_k}$.

For $c \geq 1$ we say that a Student solves (T) in c steps if the computation with any (honest) Teacher stops in at most c steps on every input $x \in \{0,1\}^n$.

Such a Student is determined by c functions

$$S_1(x), S_2(x, y_1), ..., S_c(x, y_1, ..., y_{c-1})$$

S_k computing the k-th candidate solution i_k from x and from the witnesses $y_1, ..., y_{k-1}$ received from the Teacher in earlier rounds.

The following theorem shows that the computational task (T) is in a sense hard assuming hardness of the base function f in the construction of the NW-generator. It is a simple reformulation of Theorem 2.2 from [19].

Theorem 17. *Assume that f is a function on l variables computed by poly(m)-size circuits with unique witnesses in the representation (1). Let $c \geq 1$ be a constant.*

If there are poly(m)-size formulas computing moves of Student solving (T) in c steps then there are poly(m)-size formulas computing f on at least a fraction of

$$\frac{1}{2} + \frac{1}{cm^c}$$

of all inputs.

Using Theorem 17 it is easy to obtain Theorem 16. Roughly speaking, assuming that the desired model N does not exist, the KPT theorem produces an efficeint interactive model solving (T).

5.2 Collection schema

The following argument taken from Cook and Thapen [9] shows that if a universal theory with function symbols for all functions from a natural computational class C (AC^0, NC^1 etc.) proves sharply bounded collection scheme, then functions in C are in a sense easy to invert. Therefore, provability of such a collection scheme would give us not just lower bounds on all proof systems but also an efficient algorithm for e.g. factoring (in the case of T_{PV}, C =P).

Let T be a universal theory in a language L_T containing symbols for all functions from a class C and let $f \in C$. Denote by $[x]_i$ the i-th element of the sequence coded by x. For simplicity assume that we have the function $[x]_i$ in L_T.

The collection scheme has the form

$$\forall i \leq |a| \exists x < aB(i,x) \rightarrow \exists w \forall i < |a| B(i,[w]_i)$$

where B is an open formula.

Suppose T proves the following instance of this scheme where a and y are parameters, and $m = |a|$:

$$\forall i < m \exists u < a f(u) = [y]_i \rightarrow \exists w \forall j < m f([w]_j) = [y]_j$$

This can be rewritten as

$$\exists i < m \exists w \forall u < a (f(u) = [y]_i \rightarrow \forall j < m f([w]_j) = [y]_j)$$

Applying the KPT theorem, we get functions $g_1, ..., g_k, h_1, ..., h_k \in C$ for $k \in \mathbf{N}$ such that T proves

$$\begin{aligned} \forall \overline{z} < a (f(z_1) = [y]_{g_1(y)} \rightarrow \forall j < m f([h_1(y)]_j) = [y]_j) \\ \vee (f(z_2) = [y]_{g_2(y,z_1)} \rightarrow \forall j < m f([h_2(y,z_1)]) = [y]_j) \\ ... \\ \vee (f(z_k) = [y]_{g_k(y,z_1,...,z_{k-1})} \rightarrow \forall j < m f([h_k(y,z_1,...,z_{k-1})]_j) = [y]_j) \end{aligned}$$

This gives us an algorithm which on input y (considered as a sequence $[y]_0, ..., [y]_{m-1}$), will ask for a pre-image of f on at most k elements of y and then it will output a number w coding a sequence of pre-images of all m elements y.

The algorithm is as follows. Let $w = h_1(y)$. If $\forall j < m f([w]_j) = [y]_j$ then output w and halt. Otherwise calculate $g_1(y)$ and ask for a pre-image of $[y]_{g_1(y)}$; store the answer as z_1. Then let $w = h_2(y, z_1)$. If $\forall j < m f([w]_j) = [y]_j$ then output w and halt. Otherwise calculate $g_2(y, z_1)$ and ask for a pre-image of $[y]_{g_2(y,z_1)}$; store the answer as z_2, and so on. By our assumption the algorithm will run for at most k steps of this form before it outputs a suitable w.

We can now fix a such that $|a| = m > k$, and choose a sequence $[x]_0, ..., [x]_{m-1} < a$. Let y encode the pointwise image of x under f. Run the algorithm above, and reply to queries with elements of x. We will end up with w encoding a sequence of pre-images of y. If f is not an injection and x was chosen at random, then w is probably different from x.

Thus, in a sense we can compute the inverse of f. Moreover, the resulting algorithm is still in C assuming C is a natural class like AC^0, NC^1 satisfying basic closure properties. As Cook and Thapen [9] show, this can be used e.g. to factor efficiently if $T = PV$ and C =P.

Conclusion

We surveyed some known and proved some new results concerning the unprovability of superpolynomial circuit lower bounds. Unfortunately, we did not obtain the hardness for strong systems like Frege. Nevertheless, there were many points during the exposition asking for further improvement.

For example, we saw that systems with disjoint interpolation property cannot prove superpolynomial circuit lower bounds efficiently (assuming the existence of super-bits) but it is not known whether systems like EF belong among these systems (even under any reasonable assumptions). It would be interesting to investigate also other possible modifications of interpolation (e.g. CLB-interpolation) that could hold in stronger proof systems.

The existence of super-bits implies that for any proof system P there is $f \notin$P/poly such that superpolynomial circuit lower bound for f are hard for P. Could we show that SAT$\notin$P/poly must be hard as well? As we know that the hardness of SAT$\notin$P/poly for all proof systems implies NP$\subseteq$P/poly and the existence of super-bits implies NP$\not\subseteq$P/poly, this would mean that super-bits do not exist. On the other hand, it seems plausible that the efficient provability of SAT$\notin$P/poly (maybe in Frege) implies efficient provability of $f \notin$P/poly for many f's, more precisely, an NP natural property against P/poly.

We mentioned that assuming there are BPP problems hard for NP, exponential circuit lower bounds are hard for all proof systems. Maybe it is possible to weaken the assumption to ask for the existence of BPP problems that are hard just for very specific NP algorithms and then derive the hardness of exponential circuit lower bounds, e.g. for Frege systems.

Next, we saw the gadget technique producing hard generators for all systems where PHP is hard. It seems quite achievable to improve this result to the exponential iterability of the resulting generators. Consequently we would obtain the hardness of superpolynomial circuit lower bounds for constant depth Frege systems.

We also saw the hardness of the NW-generators for a tree-like SL system. It would be very interesting to get rid of the tree-likeness condition. That is, to get the hardness for a form of Frege system working with local formulas of unbounded depth. Buss-Pudlák games give us a procedure constructing tree-like proofs from dag-like proofs. Maybe one can modify the construction in such a way that the resulting tree-like proofs have a nice structure useful for proving lower bounds.

In the last chapter we presented the unprovability of superpolynomial circuit lower bounds in theory $T_{Fla(m)}$. This could not be translated into the hardness for propositional proof systems because of the unsuitable encoding of the circuit lower bounds. Is it possible to use a specific property of VNC^1 theory which is considerably weaker than $T_{Fla(m)}$ and obtain the hardness of superpolynomial circuit lower bounds for Frege systems?

Interestingly, during the whole time we were trying more or less to prove the hardness of $f \notin$P/poly for arbitrary f. We did not use any specific properties of functions f. It is possible that for such a hardness result one needs to consider only say NP functions f (like in the speculations about super-bits). One could even

speculate how to use the uniformity property assuming that P$\neq$NP is expressible by propositional formulas.

All in all, the quest of proving the unprovability of superpolynomial circuit lower bounds offers many interesting questions and promising directions for further research.

References

[1] M. Ajtai, The complexity of the pigeonhole principle, *in: Proc. IEEE 29th Annual Symp. on Foundation of Computer Science*, (1988), pp.346-355.

[2] M. Alekhnovich, E. Ben-Sasson, A. A. Razborov, and A. Wigderson, Pseudo-random generators in propositional proof complexity, *Electronic Colloquium on Computational Complexity*, Rep. No.23, (2000). Ext. abstract in: *Proc. of the 41st Annual Symp. on Foundation of Computer Science*, (2000), pp.43-53.

[3] E. Ben-Sasson, and A. Wigderson, Short Proofs are Narrow - Resolution made Simple, *Journal of the ACM*, Vol. 48 No. 2., (2001)

[4] S. R. Buss, Bounded Arithmetic, *Naples, Bibliopolis*, (1986)

[5] S. R. Buss, Polynomial size proofs of the pigeonhole principle, *J. of Symbolic Logic*, 57, (1987), pp.916-927.

[6] S. R. Buss, and P. Pudlák, How To Lie Without Being (Easily) Convicted and Lengths of Proofs in Propositional Calculus, *In 8th Workshop on Computer Science Logic (CSL'94), Lecture Notes in Computer Science 933, Springer-Verlag*, (1995), pp.151-162.

[7] S. A. Cook, and P. Nguyen, Logical foundations of proof complexity, *Cambridge U. Press*, (2010)

[8] S. A. Cook, and Reckhow, The relative efficiency of propositional proof systems, *J. Symbolic Logic*, 44(1), (1979), pp.36-50.

[9] S. A. Cook, and N. Thapen, The strength of replacement in weak arithmetic, *ACM Translations on Computational Logic*, Vol 7:4, (2006)

[10] A. Haken, The intractability of resolution, *Theoretical Computer Science*, 39, (1985), pp.297-305.

[11] R. Impagliazzo, V. Kabanets, and A. Wigderson, In search of an easy witness: exponential time vs. probabilistic polynomial time, *Journal of Computer and System Sciences*, 65(4), (2002), pp.672-694.

[12] E. Jeřábek, Weak pigeonhole principle, and randomized computation, *Ph.D. thesis, Faculty of Mathematics and Physics, Charles U., Prague*, (2005)

[13] J.Krajíček, Lower Bounds to the Size of Constant Depth Propositional Proofs, *J. of Symbolic Logic*, 59(1), (1994), pp.73-86.

[14] J. Krajíček, Bounded arithmetic, propositional logic, and complexity theory, *Encyclopedia of Mathematics and Its Applications, Vol.60, Cambridge U. Press*, (1995)

[15] J. Krajíček, Interpolation theorems, lower bound for proof systems, and independence results for bounded arithmetic, *J. of Symbolic Logic*, 62(2), (1997), pp.457-486.

[16] J. Krajíček, On the weak pigeonhole principle, *Fundamenta Mathematicae*, Vol. 170(1-3), (2001), pp.123-140.

[17] J. Krajíček, Dual weak pigeonhole principle, pseudo-surjective functions, and provability of circuit lower bounds, *J. of Symbolic Logic*, 69(1), (2004), pp.265-286.

[18] J. Krajíček, A proof complexity generator, *in: Proc. from the 13th Int. Congress of Logic, Methodology and Philosophy of Science* (Beijing, August 2007), *King's College Publications, London, ser. Studies in Logic and Foundations of Mathematics, Eds. C.Glymour, W.Wang, and D.Westerstahl*, to app., (preprint December 2007)

[19] J. Krajíček, On the proof complexity of the Nisan-Wigderson generator based on a hard NP $\cap$ coNP function, *J. of Mathematical Logic*, to app., (preprint March 2010)

[20] J. Krajíček, Forcing with random variables and proof complexity, *London Mathematical Society Lecture Note Series, No.382, Cambridge U. Press*, (2011)

[21] J. Krajíček, and P. Pudlák, Some consequences of cryptographical conjectures for S^1_2 and EF, *Information and Computation*, 140(1), (1998), pp.82-94.

[22] J. Krajíček, P. Pudlák, and J. Sgall, Interactive Computations of Optimal Solutions, *in: B. Rovan (ed.): Mathematical Foundations of Computer Science* (B. Bystrica, August 1990), *Lecture Notes in Computer Science 452, Springer-Verlag*, (1990), pp.48-60.

[23] J. Krajíček, P. Pudlák, G. Takeuti, Bounded arithmetic and the polynomial hierarchy, *Annals of Pure and Applied Logic*, Vol. 52, (1991), pp.143-153.

[24] N. Nisan, and A. Wigderson, Hardness vs. randomness, *J. Comput. Systems Sci.*, Vol. 49, (1994), pp. 149-167.

[25] J. Pich, Nisan-Wigderson generators in proof systems with forms of interpolation, *Mathematical Logic Quarterly*, to app., (preprint March 2010)

[26] P. Pudlák, Lower bounds for resolution and cutting planes proofs and monotone computation, *J. of Symbolic Logic*, 62(3), (1997), pp.981-998.

[27] R. Raz, Resolution lower bounds for the weak pigeonhole principle, *In Proceedings of the 34th ACM Symposium on the Theory of Computing*, (2002), pp.553-562.

[28] A. A. Razborov, Bounded Arithmetic and Lower Bounds in Boolean Complexity, *Feasible Mathematics II*, (1995), pp.344-386.

[29] A. A. Razborov, Unprovability of lower bounds on the circuit size in certain fragments of bounded arithmetic, *Izvestiya of the R.A.N.*, 59(1), (1995), pp.201-224.

[30] A. A. Razborov, Resolution lower bounds for perfect matching principles, *In Proceeding of the 17th IEEE Conference on Computational Complexity*, (2002), pp.29-38.

[31] A. A. Razborov, Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution, preprint, (2003)

[32] A. A. Razborov, and S. Rudich, Natural proofs, *J.Comp. Syst. Sci*, 55(1), (1997), pp.24- 35.

[33] S. Rudich, Super-bits, demi-bits, and NP/qpoly-natural proofs, *Comp.Sci., Springer-Verlag*, Vol. 1269, (1997), pp.85-93.

[34] R. Williams, Non-Uniform ACC Circuit Lower Bounds, *26th IEEE Conference on Computational Complexity*, to app., (2011)